(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－182953
（P2002－182953A）
(43)公開日　平成14年６月28日(2002.6.28)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　12/00	５０１	Ｇ０６Ｆ　12/00	５０１Ｈ	５Ｂ０６５
	５１４		５１４Ｍ	５Ｂ０８２
3/06	３０２	3/06	３０２Ａ	５Ｄ１１０
Ｇ１１Ｂ　27/00		Ｇ１１Ｂ　27/00	Ａ	

審査請求　未請求　請求項の数２　ＯＬ　（全 13 頁）

(21)出願番号　特願2000－383124(P2000－383124)

(22)出願日　平成12年12月12日(2000.12.12)

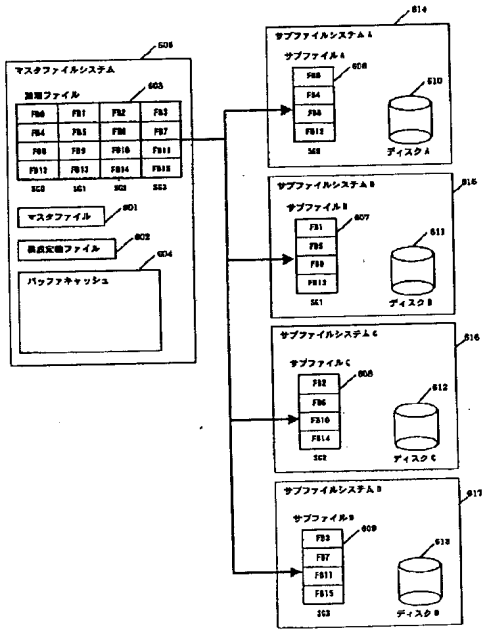
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(71)出願人　000233055
日立ソフトウエアエンジニアリング株式会社
神奈川県横浜市中区尾上町６丁目81番地
(72)発明者　西村　祐二
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】　分散ファイル管理方法

(57)【要約】
【課題】ストライピングファイル機構により複数の論理ファイルをディスク装置に分散配置する方法において、Ｉ／Ｏ要求時の並列ディスク装置アクセスの高速化を実現する。
【解決手段】論理ファイル６０３をサブファイルシステム６１４～６１７に分散配置する方法で、同一ファイルのファイルブロックＦＢ０～１５がディスク装置６１０～６１３の一括した領域に配置するためにファイルブロックの集合であるストライドグループＳＧ０～ＳＧ３を生成して、各サブファイルシステムにストライドグループ単位で一括して書き込む。また、ファイルブロックアクセス時にマスタファイルシステム６０５のバッファキャッシュ６０４に連続するファイルブロックＦＢ４～１５を先読みし、サブファイルシステムへの読み出し要求の回数を削減し、ファイルアクセスの更なる高速化を実現する。

図　6



1

【特許請求の範囲】

【請求項１】　複数ディスク装置からなるファイルシステム上で、論理ファイルを一定のサイズの複数ファイルブロックに分割し、これらのファイルブロックを複数ディスク装置に分散して配置することで、ファイル読み出し時に複数ディスク装置に並行してアクセスすることが可能なファイルシステムにおいて、ファイルブロックを分散配置することにより、物理的に一ディスク装置内で、同一ファイルのファイルブロックが不連続領域に配置されていたものを、ファイルブロックに管理情報を付与することで連続配置し、一回のＩ／Ｏ操作で読み出し／書き込みができることを特徴とする分散ファイル管理方法。

【請求項２】　請求項１項の分散ファイル管理方法において、ファイルシステム上に論理ファイルの管理情報を維持し、論理ファイルのファイルブロックに読み出し要求が生じた際、その情報に従い各ディスク装置で次に読み出すファイルブロックを予め読み出して、これらのファイルブロックをファイルシステム上のバッファキャッシュに保存することで、ディスクへの読み出し回数を削減することを特徴とする分散ファイル管理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、マスタファイルシステムに仮想的に存在する論理ファイルを、複数ディスク装置に分割して保存することで、並行して複数ディスク装置のファイルブロックにアクセス出来るようにするストライピングファイル機構と呼ばれるファイル運用方法に関するものである。

【０００２】

【従来の技術】従来より、特開平９－２２３０４９号公報に記載のような、ストライピングファイル機構による、論理ファイルを複数ディスク装置に分割して配置する方法で、論理ファイルへのＩ／Ｏ要求時に、複数ディスク装置に並行してアクセスできることは知られており、並列計算機での大容量ファイル管理において、本ファイルアクセス技術は、広く利用されている。

【０００３】

【発明が解決しようとする課題】上記従来技術は、論理ファイルを複数ディスク装置に分散配置することで、並列アクセスを可能にして、大容量ファイルへの読み出し要求の高速化を図るものであったが、ディスク装置への書き込み時、分割されたファイルブロック群に対して、同一ディスク装置へ格納するデータ群という認識をせず、各ファイルブロックを個別にディスク装置に格納するため、ディスク装置上でのファイルブロックの配置にばらつきが生じ、ファイル読み出し時、ディスク装置の回転待ち、シーク（ファイルブロックの検索）の回数が増加するため、性能劣化の原因となっていた。

【０００４】本発明の目的は、ストライピングファイル機構により論理ファイルを複数ディスク装置に分散配置する方法において、ファイル読み出し時のディスク装置へ並列での読み出しの効率化を図り、大容量ファイルへの読み出し時間の短縮をすることにある。

2

【０００５】

【課題を解決するための手段】上記目的を達成するために、ストライピングファイル機構による論理ファイルの分散管理方法において、論理ファイルを複数ディスク装置に分散配置するとき、各ディスク装置に配置する同一ファイルのファイルブロックを一括して書き込みしておくことで、ディスク装置からのファイルブロックの読み出し時にシークの回数を削減し、ファイルへのアクセス時間を短縮するものである。また、ファイルのブロックを一括してディスク装置への書き込みをすることで、ファイル読み出し時にマスタファイルシステムのバッファキャッシュに、これからアクセスが発生するファイルブロックの先読みを可能にし、各ディスク装置への読み出し要求の回数を削減するため、ファイルの読み出し時間の短縮化を図るものである。

【０００６】

【発明の実施の形態】以下、本発明の実施の形態について説明する。図１は、本発明の一実施例の全体構造であり、１０１はマスタファイル、１０２は構成定義ファイル、１０３は論理ファイル、ＦＢ０～ＦＢ１５は論理ファイルを一定のサイズに分割したファイルブロック、ＳＧ０～ＳＧ３はファイルブロックの集合であるストライドグループ、１０４は１０１、１０２、及び１０３を格納するマスタファイルシステム、１０５～１０８はサブファイルＡからＤ、１０９～１１２はディスク装置Ａからら、１１３から１１６はサブファイル、及びディスク装置を格納するサブファイルシステムＡからＤを示す。

【０００７】マスタファイルは、論理ファイルのファイルブロックがどのサブファイルに格納しているかの情報を持ち、分散配置された論理ファイルを管理する。構成定義ファイルは、ストライピングファイル機構を使用するうえで、設定するべき情報を定義する。これらの情報には、ストライピングファイル機構が使用するサブファイルシステム、ファイルを分割する際のファイルブロックのサイズが含まれる。論理ファイルは、分割して管理する対象となるファイルで、マスタファイルシステム内に論理上作成される。ファイルブロックは、論理ファイルを一定のサイズに分割したファイルブロックで、サブファイルとして分散管理される最小単位である。ストライドグループは、同一ファイルシステムに分配されるファイルブロックをグループ化したもので、サブファイル分割のときに用いる単位である。マスタファイルシステムは、ストライピングファイル機構を使用するファイルシステムで、論理ファイル、マスタファイルを管理するファイルシステムである。各サブファイルは、サブファイルシステム上の実データが格納されているファイルで

ある。各ディスク装置は、物理的にファイルを格納する装置で、１サブファイルシステムは、１ディスク装置をマウントすることにより、ファイルシステムを作成する。各サブファイルシステムは、サブファイルが格納されているファイルシステムである。

【０００８】本発明は、マスタファイルシステムにある論理ファイルを一定のサイズに分割したファイルブロックを、複数のサブファイルシステムのディスク装置に分割して配置し、アプリケーションからのＩ／Ｏ要求をそれぞれのサブファイルが並行して実行することで、ファイルアクセスの高速化をはかるものである。図１で、論理ファイルは、一定のサイズに分割されたファイルブロック０から１５で構成している。これらのファイルブロックを複数のディスク装置に配置するマスタファイルシステムは、その構成定義ファイルによりこれらファイルブロックの分配対象となるサブファイルシステムを予め限定している。図１で、サブファイルシステム用にマウントされたディスク装置が４個であるとする。このとき、論理ファイルのファイルブロック０、４、８、１２がサブファイルシステムＡのディスク装置Ａへ、ファイルブロック１、５、９、１３がサブファイルシステムＢのディスク装置Ｂへ、ファイルブロック２、６、１０、１４がサブファイルシステムＣのディスク装置Ｃへ、ファイルブロック３、７、１１、１５がサブファイルシステムＤのディスク装置Ｄへ分散配置することは、構成定義ファイルから導かれる対象サブファイルシステムとファイルブロック数により決定している。論理ファイルとサブファイルとの関係を管理するマスタファイルは、ファイルブロックをディスク装置に配置するとき、ファイルブロック単位で当該ディスク装置に書き込みをすることなく、最初に同一ディスク装置に分配されるファイルブロックをグループ化する。この一纏にしたファイルブロック群をストライドグループと呼ぶ。ファイルブロック０、４、８、１２をストライドグループ０、ファイルブロック１、５、９、１３をストライドグループ１、ファイルブロック２、６、１０、１４をストライドグループ２、ファイルブロック３、７、１１、１５をストライドグループ３として、各ストライドグループ単位で対象サブファイルシステムのディスク装置へ一括して書き込む。

【０００９】図２に、本方法における一ファイル分配処理の詳細を説明する。図２は、本分散ファイル管理方法を実現するために、マスタファイルシステムが論理ファイルの分割時に行う処理について示すものである。Ｓ２０１で、構成定義ファイルより、ストライピングファイル機構に使用可能なサブファイルシステムとその数を、またファイルを分割するファイルブロックのサイズを読み出す。Ｓ２０２で論理ファイルをファイルブロックサイズに分割し、分割したファイルブロックの総数をｎ個とする。分割したファイルブロックを前から順番にブロック番号０からｎ－１とする。このとき、最後のファイルブロックが定義されたファイルブロックサイズに満たないときでも、１ファイルブロックとみなし、ブロック番号を定義する。ファイルブロックにブロック番号を定義した後、Ｓ２０３で、ブロック番号をサブファイルシステム数で割った余りが同じファイルブロックを集めてストライドグループにする。また、この余りをストライドグループ番号とする。Ｓ２０４において、Ｓ２０３で生成したストライドグループをストライドグループ番号の順にサブファイルシステムに割り当てる。

【００１０】図３では、上記分散ファイル管理方法を用いて、複数の論理ファイルを同一ディスク装置に分配したときのファイルブロックの配置を説明する。図３において、３０１はマスタファイル、３０２は構成定義ファイル、３０３は論理ファイル１、３０４は論理ファイル２、ＦＢ０～ＦＢ１５は論理ファイル１を一定のサイズに分割したファイルブロック、ＳＧ０～ＳＧ３は論理ファイル１のファイルブロックの集合であるストライドグループ、ＦＢ１６～ＦＢ２７は論理ファイル２を一定のサイズに分割したファイルブロック、Ｓｇ０～Ｓｇ３は論理ファイル２のファイルブロックの集合であるストライドグループ、３０５は３０１、３０２、３０３、及び３０４を格納するマスタファイルシステム、３０６～３０９は論理ファイル１を分割しサブファイルシステムに配置したサブファイル１Ａ～１Ｄ、３１０～３１３は論理ファイル２を分割しサブファイルシステムに配置したサブファイル２Ａ～２Ｄ、３１４～３１７はディスク装置ＡからＤ、３１８～３２１は二つのサブファイル、及びディスク装置を格納するサブファイルシステムＡ～Ｄを示す。サブファイルシステムＡのディスク装置Ａにおいて、論理ファイル１のストライドグループＳＧ０と論理ファイル２のストライドグループＳｇ０の書き込みが競合したとき、それぞれのファイルブロック群がストライドグループとして一括してディスク装置Ａに書き込まれる。これにより、２つの異なる論理ファイルに属するファイルブロックが交互に（または、混じり合って）ディスク装置に書き込まれることを避け、ディスク装置内の空き領域になるべく同一論理ファイルのファイルブロックが続けて配置されることで、ファイルブロックの読み出し時の回転待ち、シークを削減することが可能となり、ファイルの読み出し要求の効率化を図る。

【００１１】図４に、サブファイルシステムにおいて、マスタファイルシステムより割り当てられたストライドグループをディスク装置に配置する処理の方法を説明する。Ｓ４０１で、マスタファイルシステムよりストライドグループ単位のファイルブロック群を受け取る。Ｓ４０２において、マスタファイルシステムから受け取ったファイルブロックを一括して、ディスク装置の空き領域に割り当てていく。本処理方式により、２論理ファイル以上のファイルブロック群を同一ディスク装置に書き込

5

もうとすると、必ず同一論理ファイルのファイルブロック群に対する書き込みが完了してから、次の論理ファイルのファイルブロック群に対する書き込みが開始されるため、異なる論理ファイルに属するファイルブロックが混在して、ディスク装置に書き込まれることが無いことを保証する。

【００１２】次に本分散ファイル管理方法において、論理ファイルへの読み出し時のファイルアクセス方式について説明する。図５は、図１における論理ファイルに読み出し要求があった場合のファイルアクセス方法を説明するもので、５０１は論理ファイル、ＦＢ０～ＦＢ１５は論理ファイルを一定のサイズに分割したファイルブロック、ＳＧ０～ＳＧ３はファイルブロックの集合であるストライドグループ、５０２～５０５はサブファイルＡからＤ、５０６～５０９はディスク装置ＡからＤを示す。図５で、論理ファイルのファイルブロック０から４に読み出し要求があった場合、ファイルブロック０から３までを並行に読み出すことが可能で、且つ、ディスク装置Ａではファイルブロック０の読み出し要求を実行した後、直ちにファイルブロック４の読み出し要求を実行することが出来る。従来は、ファイルへの書き込み時、分割されたファイルブロック群に対して、同一ディスク装置へ格納すると認識せずに、各ブロックを別々にディスク装置に書き込んでいたため、ファイル読み出し時にディスク装置上でのファイルブロックの配置にばらつきが生じ、ディスク装置の回転待ち、シークの回数が増加し、性能劣化の原因となった。一方、同じ論理ファイルに属するファイルブロックを一纏めにして書き込んだ場合、読み出し時の回転待ち、シークの回数を削減し、効率よく同一論理ファイルのファイルブロック群を読み出すことが可能である。

【００１３】次に示す実施例では、本ファイル管理方法において、サブファイルを読み出すとき、マスタファイルが管理している情報より、ファイルブロックを先読みし、それらをマスタファイルシステムのバッファキャッシュに保存する方法を示す。本ファイル管理方法では読み出し要求があった場合、最初にアクセスがあったファイルブロックより、同じディスク装置で次に読み出し要求が起こり得るファイルブロックは現在読み出しているファイルブロックと一括して書き込んでいるため、ファイルブロックを先読みして、マスタファイルシステムのバッファキャッシュに保存することが出来る。

【００１４】図６は、本分散ファイル管理方法においてマスタファイルシステムにバッファキャッシュを装備したものを説明する図で、６０１はマスタファイル、６０２は構成定義ファイル、６０３は論理ファイル、６０４はバッファキャッシュ、ＦＢ０～ＦＢ１５は論理ファイルを一定のサイズに分割したファイルブロック、ＳＧ０～ＳＧ３はファイルブロックの集合であるストライドグループ、６０５は６０１、６０２、６０３、及び６０４

6

を格納するマスタファイルシステム、６０６～６０９はサブファイルＡからＤ、６１０～６１３はディスク装置ＡからＤ、６１４～６１７はサブファイル及びディスク装置を格納するサブファイルシステムＡからＤを示す。

【００１５】バッファキャッシュは、マスタファイルシステムに装備し、サブファイルシステムより先読みしたファイルブロックを格納する領域として使用する。

【００１６】図６において、サブファイルＡのファイルブロック０にアクセスすると次にファイルブロック４、８、及び１２にアクセスされることが起こり得るため、これらの領域をファイルブロック０の読み出し要求が出た時点で、マスタファイルシステム上のバッファキャッシュに先読みすることで、再度マスタファイルシステムからサブファイルシステムへ読み出し要求を出すことなく、直接マスタファイルシステムが論理ファイルの読み出し要求に応えることが可能となる。このため、さらにファイルアクセスの高速化が図ることができる。

【００１７】図７では、バッファキャッシュを装備したマスタファイルシステムを用い、読み出し要求があった場合のファイルアクセス方法を説明するもので、７０１は論理ファイル、７０２はバッファキャッシュ、ＦＢ０～ＦＢ１５は論理ファイルを一定のサイズに分割したファイルブロック、ＳＧ０～ＳＧ３はファイルブロックの集合であるストライドグループ、７０３～７０６はサブファイルＡからＤ、７０７～７１０はディスク装置ＡからＤを示す。図７で、論理ファイルのファイルブロック０から８に読み出し要求があった場合、本方式では、ファイルブロック０に読み出し要求が発生すると、ファイルブロック０が属すストライドグループをバッファキャッシュにアップロードすることで、ファイルブロック４、８、１２に読み出し要求する際にサブファイルシステムＡのディスク装置Ａにアクセスする必要性が無くなり、代わりにバッファキャッシュにあるファイルブロックにアクセスすることが可能となる。同様にストライドグループＳＧ１、ＳＧ２、及びＳＧ３についてもバッファキャッシュにアップロードすることが出来る。このため、論理ファイルへの読み出し要求の範囲が大きいとき、ストライピングファイル機構を使って複数ディスク装置から並行してファイルブロックを読み出すとともに、１回のディスク装置への読み出し要求でディスク装置内に存在する後続のファイルブロックをアップロードできるため、以後サブファイルシステムに読み出し要求を出す必要がなくなり、ファイルアクセスの高速化が図ることが出来る。

【００１８】図８、９に、本ファイルアクセス方法でバッファキャッシュを装備したときの処理について、フローチャートを用いて説明する。図８のフローチャートはマスタファイルシステムでの処理を、図９のフローチャートは各サブファイルシステムでの処理を示す。

【００１９】図８の処理において、マスタファイルシス

7

テムはＳ８０１でアプリケーションより読み出し要求を受けると、Ｓ８０２の読み出し要求で、最初のファイルブロックにアクセスをする。このとき、Ｓ８０３で、まず読み出し要求が発生したファイルブロックが、マスタファイルシステムのバッファキャッシュ内に存在するかを確認する。最初にアクセスしたファイルブロックは、キャッシュ内に存在しないため、Ｓ８０５に移行する。Ｓ８０５では、マスタファイルが管理している情報より、配置したサブファイルシステムに対して、当該ファイルブロックと後続するファイルブロックの読み出し要求を出す。２つ目以降のファイルブロックのアクセスでは、当該ファイルブロックがバッファキャッシュに格納されている場合もあるため、その場合はＳ８０３からＳ８０４に移行する。Ｓ８０４では、マスタファイルシステムのバッファキャッシュより読み出し要求のあったファイルブロックを読み出す。Ｓ８０６で、読み出し要求のあった全てのファイルブロックへのアクセスが完了したかを確認する。読み出し要求への読み出しが完了している場合、処理を終了する。読み出し要求に対する読み出しが完了していない場合、Ｓ８０７で読み出し要求の出ている次のファイルブロックにアクセスをする。このとき、ファイルブロックが、バッファキャッシュ内に存在しているかを調べるため、Ｓ８０３に戻る。このように、マスタファイルシステムでは、順次読み出し要求のあったファイルブロックにアクセスをして、バッファキャッシュに当該ファイルブロックが存在する場合、バッファキャッシュのファイルブロックを用い読み出し要求に応え、バッファキャッシュに無い場合、当該ファイルブロックを配置しているサブファイルシステムに読み出し要求を出す。

【００２０】図９で、バッファキャッシュを装備する方法において、サブファイルシステム側の処理を説明する。まず、Ｓ９０１でマスタファイルシステムから読み出し要求を受け取り、Ｓ９０２で読み出し要求のあったファイルブロックから読み出し、マスタファイルシステムにファイルブロックを送信する。Ｓ９０３において、読み出し要求のあった同一ストライドグループのファイルブロックがあると、そのブロックに対しての読み出すためＳ９０２の処理を繰り返す。Ｓ９０３において、同一ストライドグループに属する全てのファイルブロックの読み出しが完了していると処理が終了する。このとき、本方法では、ファイル書き込み時にストライドグループを用い、同一論理ファイルのファイルブロックを一括して当該ディスク装置の連続した領域に配置しているため、読み出し要求時、ファイルブロックを一括して読み出すことが可能となる。

【００２１】

8

【発明の効果】本発明では、複数の論理ファイルのファイルブロックが同一ディスク装置に配置される場合、これらのファイルブロックをディスク装置内で混在して配置することを回避し、それぞれのファイルブロックを一括して書き込むことで、論理ファイルの読み出し時、読み出し要求のあったファイルブロックを検索する時間の短縮を可能にする。また、ファイルブロックの先読みをすることで、マスタファイルシステムからサブファイルシステムへの読み出し要求の回数を削減することができ、大容量ファイルアクセス時間の短縮を実現する。

【図面の簡単な説明】

【図１】本発明の一実施例のマスタファイルシステムとサブファイルシステムの構成を示す全体図である。

【図２】図１の実施例において、論理ファイルの分割方法の処理を示すフローチャートである。

【図３】図１の実施例において、複数の論理ファイルが同一サブファイルシステムを共有したときを示す図である。

【図４】図１の実施例において、書き込みに対してサブファイルシステム側でのファイルブロックの書き込み処理を示すフローチャートである。

【図５】図１の実施例において、ストライドグループがそれぞれのサブファイルシステムに分割された後、論理ファイルに読み出し要求があったときの、ファイルアクセス方法を示す図である。

【図６】本発明の実施例である分散ファイル管理方式で、マスタファイルシステムにバッファキャッシュを装備して、ファイル読み出しのあったファイルブロックと同一のストライドグループに属するファイルブロックを、マスタファイルシステムに先読みする方法を示す全体構成図である。

【図７】図６のバッファキャッシュを装備する実施例で、論理ファイルに読み出し要求があったときの、ファイルアクセス方法とファイルブロックの先読み方法の処理を示す図である。

【図８】図６のファイルブロックを先読みする実施例で、マスタファイルシステムに読み出し要求があった場合の処理を示すフローチャートである。

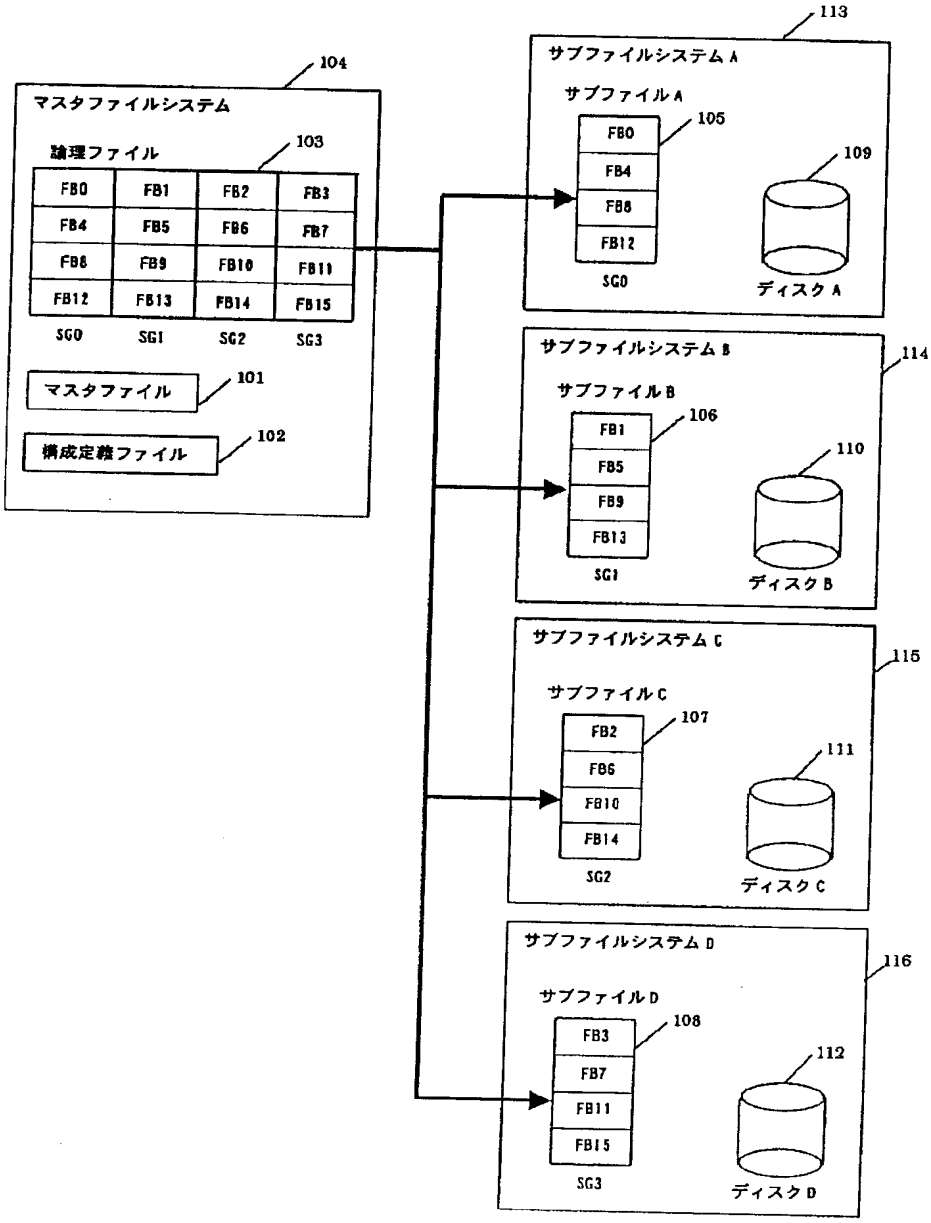
【図９】図６のファイルブロックを先読みする実施例で、サブファイルシステムがマスタファイルシステムから読み出し要求を受けたときの処理を示すフローチャートである。

【符号の説明】

６０２…構成定義ファイル、６０３…論理ファイル、６０４…バッファキャッシュ、６０５…マスタファイルシステム、６０６…サブファイルＡ、６１０…ディスク装置Ａ、６１４…サブファイルシステムＡ。

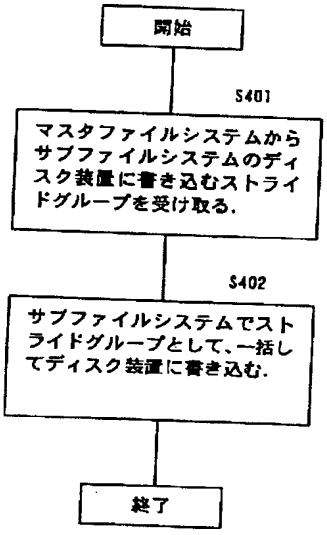
【図１】

図　1


104
マスタファイルシステム
論理ファイル
103
FB0 FB1 FB2 FB3
FB4 FB5 FB6 FB7
FB8 FB9 FB10 FB11
FB12 FB13 FB14 FB15
SG0 SG1 SG2 SG3
マスタファイル
101
構成定義ファイル
102
113
サブファイルシステム A
サブファイル A
FB0
FB4
FB8
FB12
105
109
SG0
ディスク A
サブファイルシステム B
114
サブファイル B
FB1
FB5
FB9
FB13
106
110
SG1
ディスク B
サブファイルシステム C
115
サブファイル C
FB2
FB6
FB10
FB14
107
111
SG2
ディスク C
サブファイルシステム D
116
サブファイル D
FB3
FB7
FB11
FB15
108
112
SG3
ディスク D

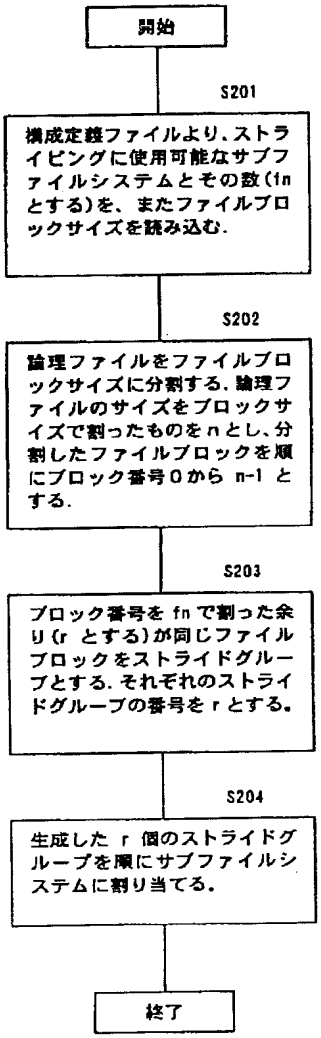
【図４】

図　4


開始
S401
マスタファイルシステムからサブファイルシステムのディスク装置に書き込むストライドグループを受け取る.
S402
サブファイルシステムでストライドグループとして、一括してディスク装置に書き込む.
終了

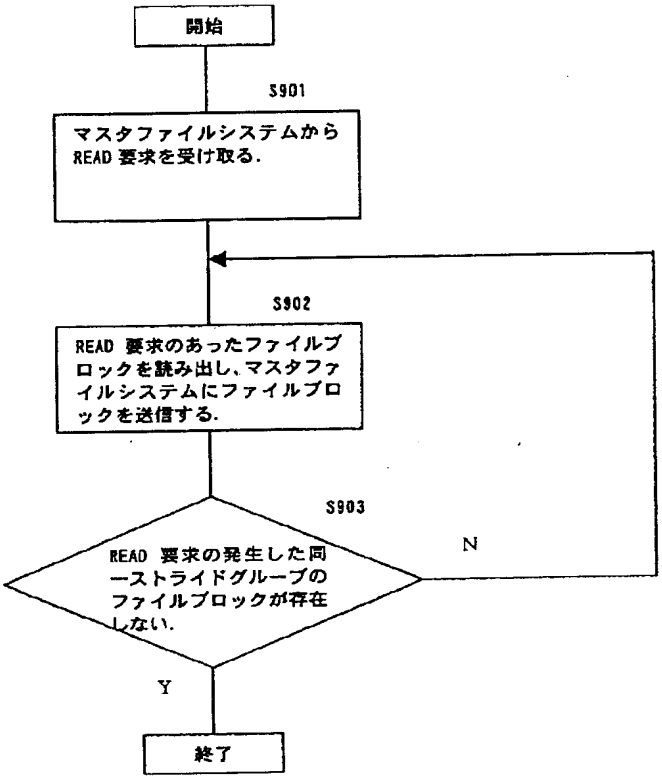
【図２】

図　2


開始
S201
構成定義ファイルより、ストライピングに使用可能なサブファイルシステムとその数(fnとする)を、またファイルブロックサイズを読み込む.
S202
論理ファイルをファイルブロックサイズに分割する. 論理ファイルのサイズをブロックサイズで割ったものをnとし、分割したファイルブロックを順にブロック番号0から n-1 とする.
S203
ブロック番号を fn で割った余り(r とする)が同じファイルブロックをストライドグループとする. それぞれのストライドグループの番号を r とする。
S204
生成した r 個のストライドグループを順にサブファイルシステムに割り当てる.
終了

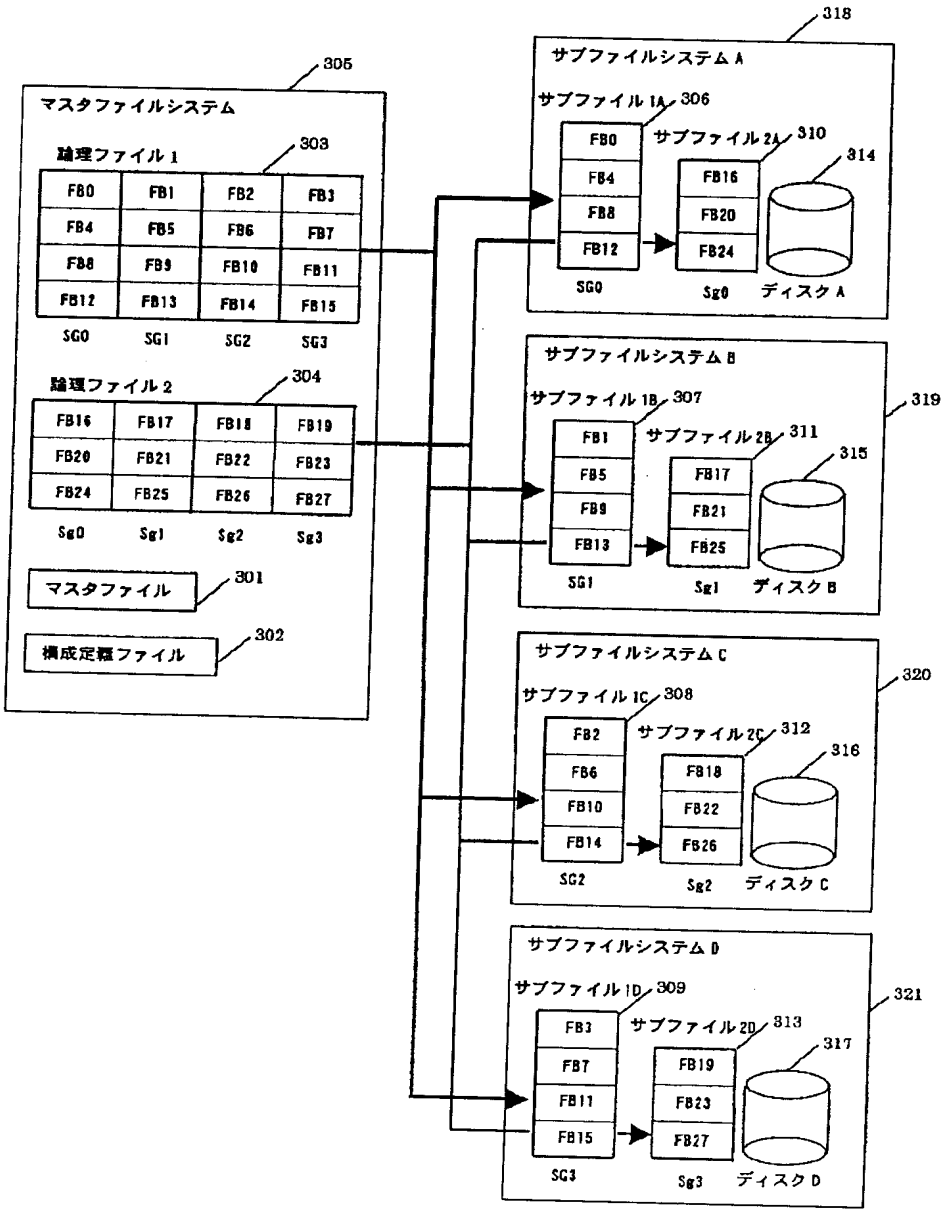
【図９】

図　9


開始
S901
マスタファイルシステムからREAD 要求を受け取る.
S902
READ 要求のあったファイルブロックを読み出し、マスタファイルシステムにファイルブロックを送信する.
S903
READ 要求の発生した同一ストライドグループのファイルブロックが存在しない.
N
Y
終了

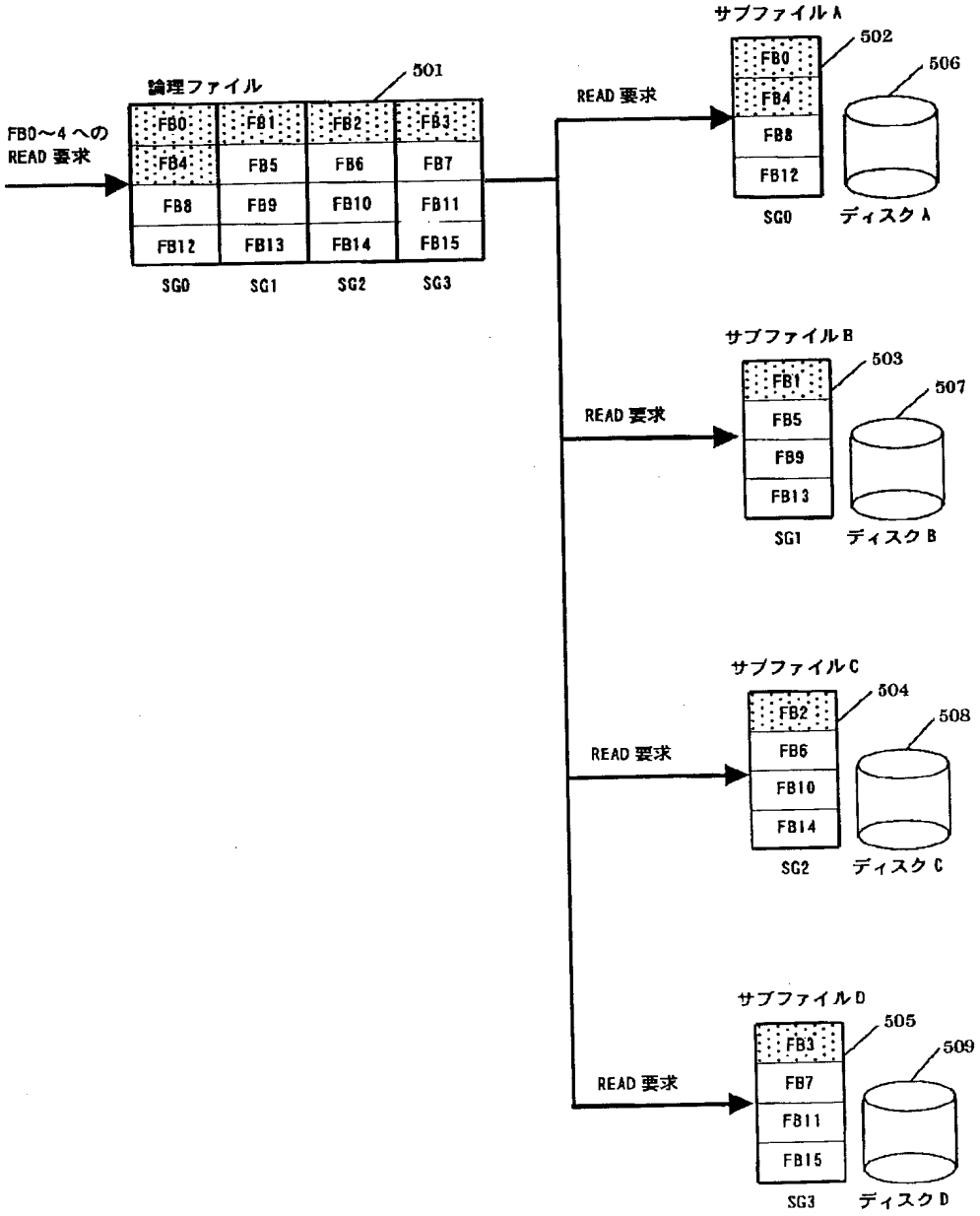
【図３】

図　3


305
マスタファイルシステム
論理ファイル１
303
FB0 FB1 FB2 FB3
FB4 FB5 FB6 FB7
FB8 FB9 FB10 FB11
FB12 FB13 FB14 FB15
SG0 SG1 SG2 SG3
論理ファイル２
304
FB16 FB17 FB18 FB19
FB20 FB21 FB22 FB23
FB24 FB25 FB26 FB27
Sg0 Sg1 Sg2 Sg3
マスタファイル
301
構成定義ファイル
302
318
サブファイルシステムＡ
サブファイル１A
306
FB0
FB4
FB8
FB12
SG0
サブファイル２A
310
FB16
FB20
FB24
Sg0
314
ディスクＡ
サブファイルシステムＢ
319
サブファイル１B
307
FB1
FB5
FB9
FB13
SG1
サブファイル２B
311
FB17
FB21
FB25
Sg1
315
ディスクＢ
サブファイルシステムＣ
320
サブファイル１C
308
FB2
FB6
FB10
FB14
SG2
サブファイル２C
312
FB18
FB22
FB26
Sg2
316
ディスクＣ
サブファイルシステムＤ
321
サブファイル１D
309
FB3
FB7
FB11
FB15
SG3
サブファイル２D
313
FB19
FB23
FB27
Sg3
317
ディスクＤ

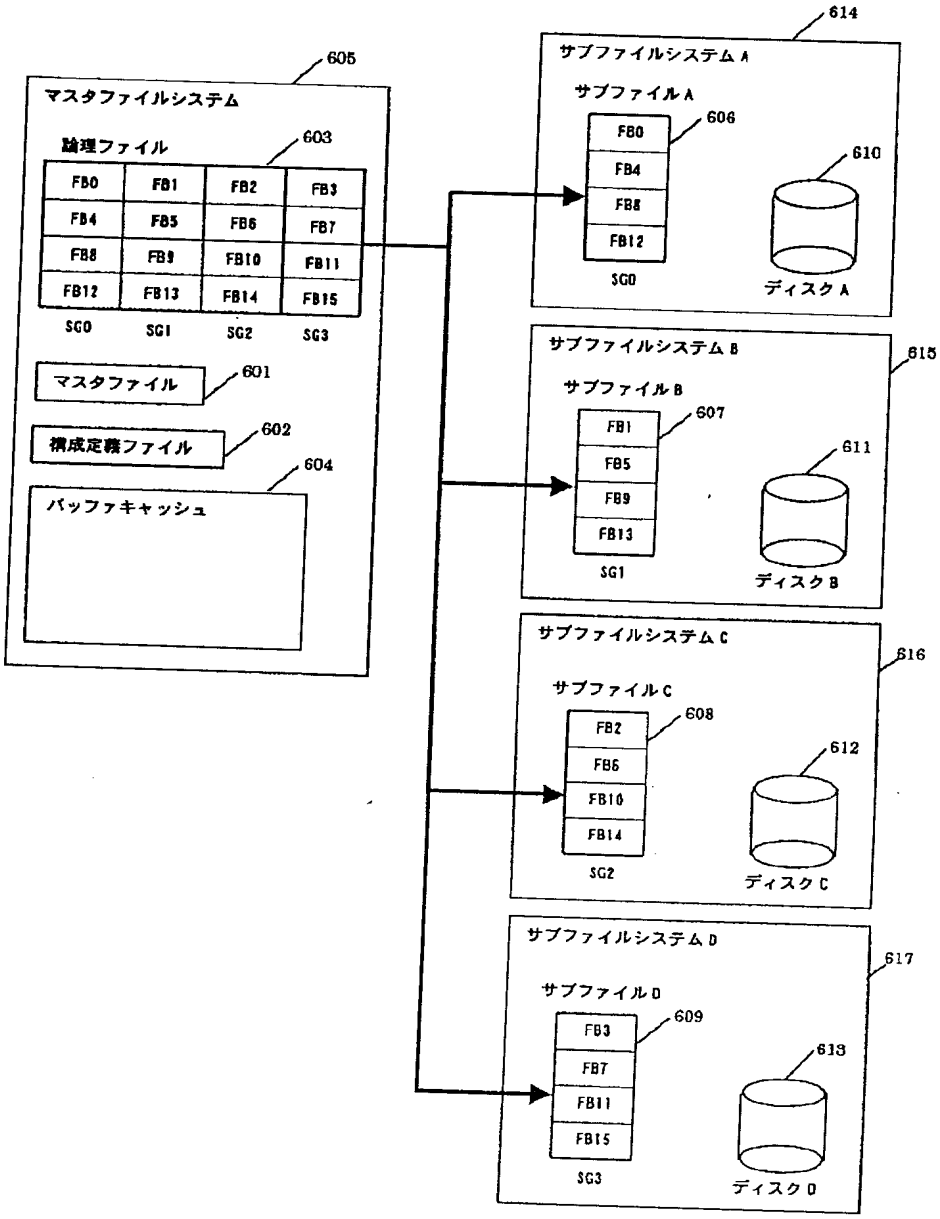
【図５】

図　5


論理ファイル
501
FB0
FB1
FB2
FB3
FB4
FB5
FB6
FB7
FB8
FB9
FB10
FB11
FB12
FB13
FB14
FB15
SG0
SG1
SG2
SG3
FB0～4 への
READ 要求
READ 要求
READ 要求
READ 要求
READ 要求
サブファイル A
502
506
FB0
FB4
FB8
FB12
SG0
ディスク A
サブファイル B
503
507
FB1
FB5
FB9
FB13
SG1
ディスク B
サブファイル C
504
508
FB2
FB6
FB10
FB14
SG2
ディスク C
サブファイル D
505
509
FB3
FB7
FB11
FB15
SG3
ディスク D

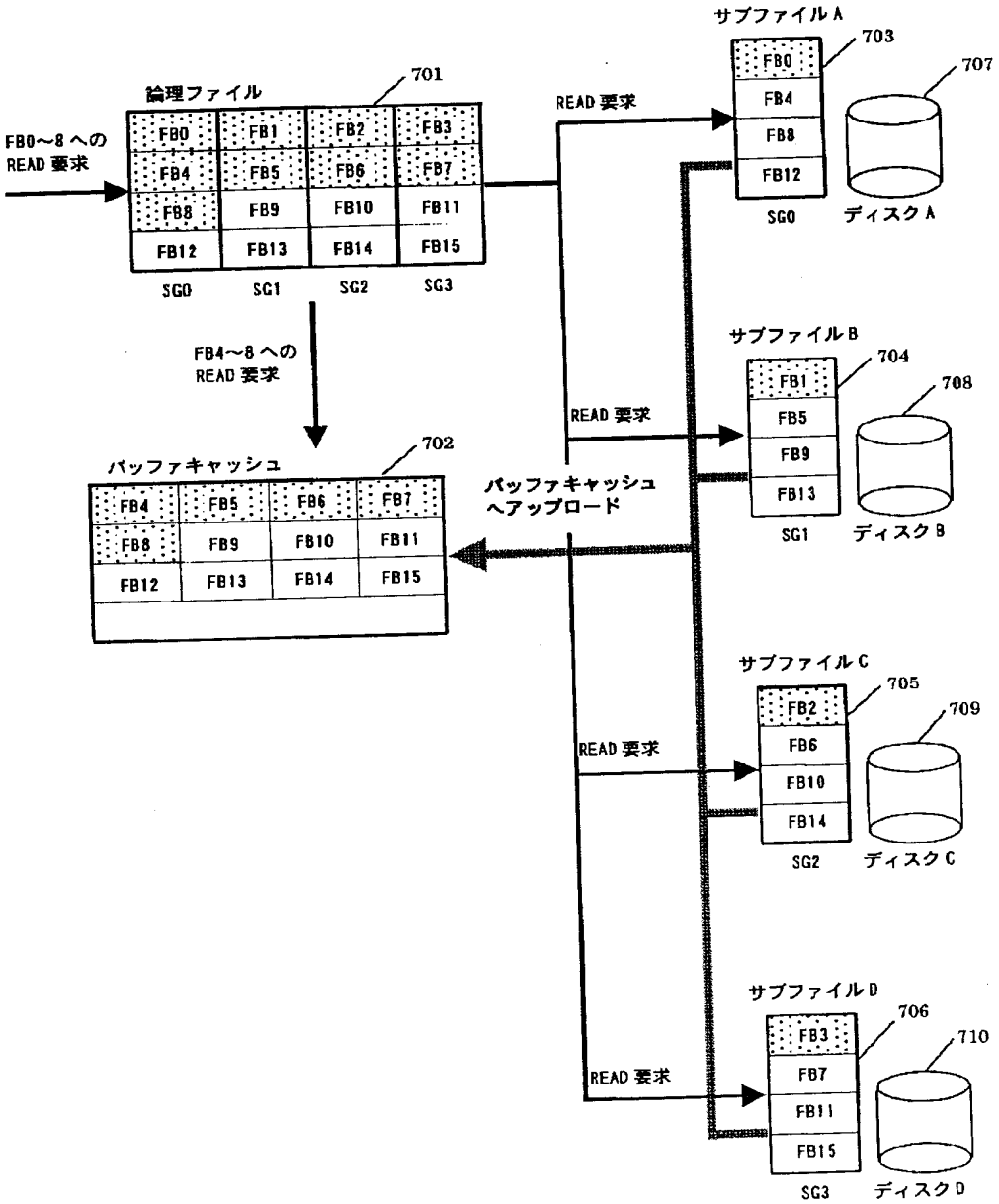
【図６】

図　6


マスタファイルシステム
605
論理ファイル
603
FB0 FB1 FB2 FB3
FB4 FB5 FB6 FB7
FB8 FB9 FB10 FB11
FB12 FB13 FB14 FB15
SG0 SG1 SG2 SG3
マスタファイル
601
構成定義ファイル
602
604
バッファキャッシュ
サブファイルシステム A
614
サブファイル A
FB0
FB4
FB8
FB12
606
SG0
610
ディスク A
サブファイルシステム B
615
サブファイル B
FB1
FB5
FB9
FB13
607
SG1
611
ディスク B
サブファイルシステム C
616
サブファイル C
FB2
FB6
FB10
FB14
608
SG2
612
ディスク C
サブファイルシステム D
617
サブファイル D
FB3
FB7
FB11
FB15
609
SG3
613
ディスク D

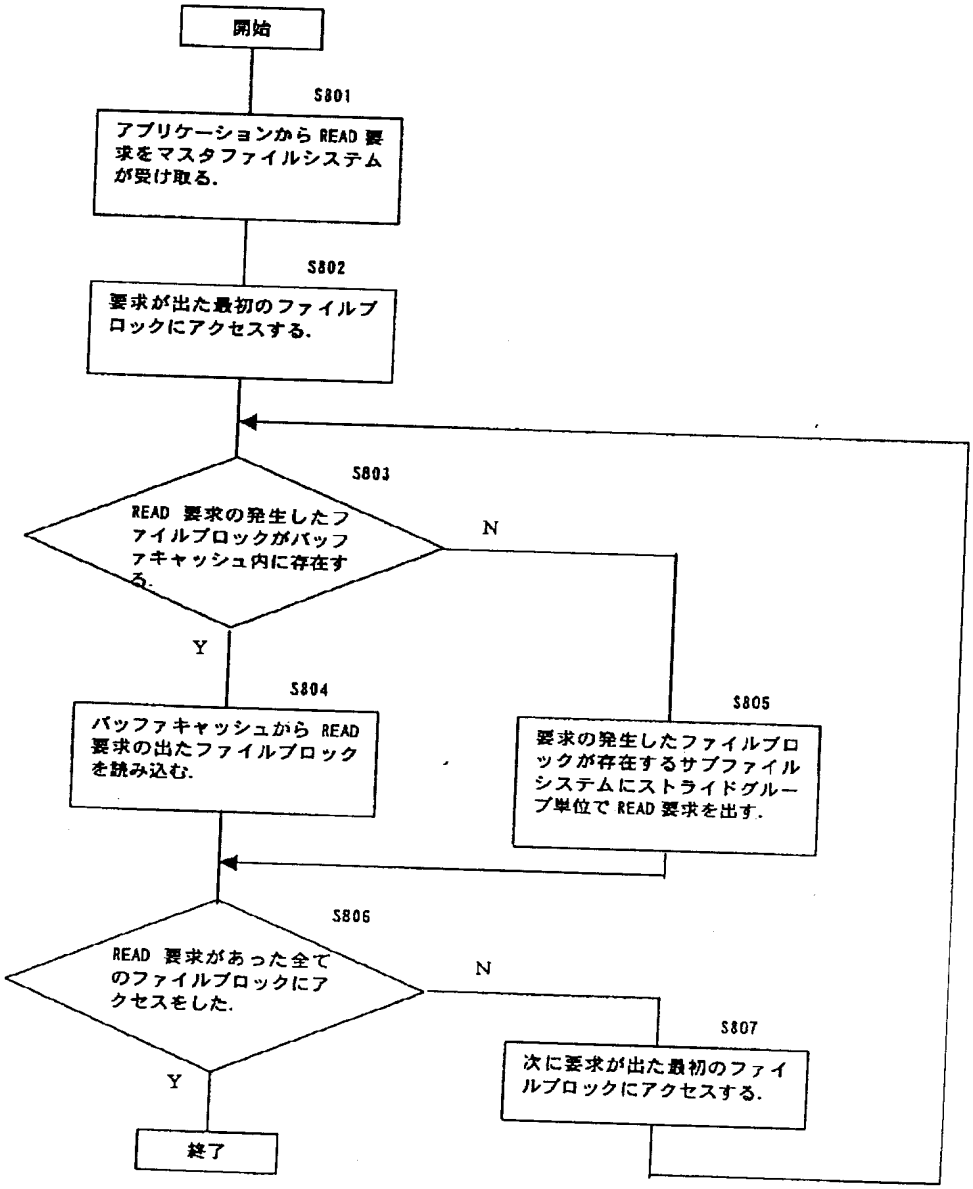
【図７】

図　7


論理ファイル
701
FB0
FB1
FB2
FB3
FB4
FB5
FB6
FB7
FB8
FB9
FB10
FB11
FB12
FB13
FB14
FB15
SG0
SG1
SG2
SG3
FB0～8 への
READ 要求
FB4～8 への
READ 要求
バッファキャッシュ
702
バッファキャッシュ
へアップロード
READ 要求
READ 要求
READ 要求
READ 要求
サブファイル A
703
707
SG0
ディスク A
サブファイル B
704
708
SG1
ディスク B
サブファイル C
705
709
SG2
ディスク C
サブファイル D
706
710
SG3
ディスク D

【図８】

図　８



フロントページの続き

(72)発明者　杉山　和仁
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内

(72)発明者　松井　和吉
神奈川県横浜市中区尾上町六丁目81番地　日立ソフトウェアエンジニアリング株式会社内

Ｆターム(参考) 5B065 BA01 CH02
5B082 CA18 CA20 FA12
5D110 AA13 DA11 DB05 DC03 DC16
DE01

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-182953
(43)Date of publication of application : 28.06.2002

(51)Int.Cl. G06F 12/00
G06F 3/06
G11B 27/00

(21)Application number : 2000-383124
(22)Date of filing : 12.12.2000

(71)Applicant : HITACHI LTD
HITACHI SOFTWARE ENG CO LTD
(72)Inventor : NISHIMURA YUJI
SUGIYAMA KAZUHITO
MATSUI WAKICHI

(54) DISTRIBUTED FILE MANAGEMENT METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To accelerate parallel disk device access at the time of I/O requests in a method of distributing and arranging a plurality of logic files to disk units by a striping file mechanism.
SOLUTION: In this method of distributing and arranging the logic file 603 to sub-filing systems 614-617, stride groups SG0-SG3 which are the sets of file blocks are generated so as to arrange the file blocks FB0-FB15 of the same file in the batch area of the disk devices 610-613 and they are written altogether to the respective sub-filing systems by the stride group unit. Also, when accessing the file block, the successive file blocks FB4-FB15 are looked ahead to the buffer cache 604 of a master filing system 605, the number of times of requesting read to the sub-filing systems is reduced and file access is accelerated further.

図 5